Vers des machines exaflopiques vertes

Mohammed Diouri, Olivier Glück, Laurent Lefevre

RenPar'20 / SympA'14 / CFSE 8
Saint-Malo, France, du 10 au 13 mai 2011

Vers des machines exaflopiques vertes

Mohammed El Mehdi Diouri, Olivier Glück et Laurent Lefèvre

Equipe RESO, Laboratoire LIP
UMR 5668 (CNRS, ENS, INRIA, UCB)
46 Allée d'Italie 69364 Lyon - France
{Mehdi.Diouri, Olivier.Gluck, Laurent.Lefevre}@ens-lyon.fr

Résumé
Les superordinateurs ont connu une croissance rapide en particulier ces dernières années. À peine avons-nous franchi l'échelle du petaflop que l'on s'intéresse déjà à concevoir des machines pouvant atteindre 1 exaflop par seconde et ce, afin d'être en mesure de satisfaire les besoins importants en terme de performances qu'expriment les scientifiques dans divers domaines. Cependant, pour concevoir des machines exaflopiques, il faut au préalable relever certains défis dont le principal est d'être capable de réduire les importants coûts de consommations énergétiques dans ces machines. Dans cet article, nous nous demandons si les solutions existantes pour réduire la consommation énergétique au niveau pétaflopique resteront valables à l'échelle exaflopique et dans quelles mesures il est possible de les adapter pour qu'elles puissent passer à l'échelle. Nous proposons également des solutions nouvelles pour l'échelle exaflopique, qui débouchent sur une architecture verte pour les machines exaflopiques.

Mots-clés : Machines exaflopiques, Efficacité énergétique, Calcul haute performance, Superordinateurs

1. Introduction

Un superordinateur est une machine construite à partir d'une collection d'ordinateurs capables d'effectuer des tâches en parallèle et permet ainsi d'atteindre des performances très élevées. Depuis le début des années 90, ces superordinateurs ont connu une croissance rapide ; on peut constater grâce à la liste Top500 [1] qu'environ tous les 11 ans, les performances des superordinateurs connaissent une croissance d'un facteur 1000. 1 gigaflop a été atteint en 1985 par le Cray 2 [2], 1 téraflop a été atteint en 1997 par le système ASCI Red [3], 1 pétaflop a été atteint en 2008 par Roadrunner [4]. Selon la liste Top500 publiée en novembre 2010, le superordinateur le plus puissant est Tianhe-1A [5] une machine comportant plus de 180 000 coeurs et capable d'atteindre plus que 2,5 pétaflops. Cette tendance linéaire de la croissance de la performance n'a pas cessé et est loin de connaître un quelconque ralentissement vu qu'il est prévu d'atteindre 10 pétaflops en 2011 avec Bluewaters [6], et de franchir le seuil des 20 pétaflops d'ici 2012 avec la machine Sequoia [14]. Si on s'appuie sur ces projections linéaires de performance, on ne peut s'empêcher de penser qu'on est sur la bonne voie pour franchir le seuil de l'ère exaflopique d'ici 2018.
Une machine exaflopique est un superordinateur capable d'effectuer plus que 10^{18} opérations à virgule flottante par seconde. Pour atteindre de telles performances, une machine exaflopique devrait par exemple associer plus que de 166 millions de coeurs répartis dans 583 racks composés chacun de 384 noeuds à 1.5 GHz [15].
Le besoin d'aller au plus vite vers cette ère exafloplique est d'autant plus exprimé que l'on sait qu'à l'heure actuelle, certains problèmes complexes sont impossibles à traiter dans un laps de temps raisonnable avec des machines pétaflopiques et que seule l'émergence de machines exaflopiques permettra de les résoudre. La nécessité de résoudre ces problèmes très complexes est notamment éprouvée :

1. Liste Top500 : `http://www.top500.org/`
2. Cray 2 : `http://archive.computerhistory.org/resources/text/Cray/Cray.Cray2.1985.102646185.pdf`
3. ASCI Red : `http://www.sandia.gov/ASCI/Red/index.html`
4. Roadrunner : `http://www.lanl.gov/roadrunner/`
5. Tianhe-1A in TOP500 Supercomputing Sites : http ://www.top500.org/system/10587
6. BlueWaters project : `http://www.ncsa.illinois.edu/BlueWaters/`

– en climatologie [7], pour être capable de prédire les phénomènes climatiques extrêmes tels que les canicules, sécheresses, et inondations ;
– en médecine, où la possibilité d'effectuer le calcul complexe du génome humain d'un patient rendrait possible la prescription de traitements individualisés ;
– dans le domaine de l'énergie nucléaire, pour permettre d'analyser les menaces et de prédire les effets des explosions nucléaires [11] ;
– mais aussi dans d'autres domaines, tel que les nanosciences, la sismologie, la chimie, etc.

Cependant, pour assurer le passage à l'ère exaflopique, il faudra être capable de relever plusieurs défis qui deviendront encore plus problématiques à cette échelle. Parmi ces défis, on retrouve notamment les questions relatives à :

– la fiabilité et la tolérance aux fautes : selon [1], les machines exaflopiques connaîtront divers types de fautes de manière continue ;
– la parallélisation à très large échelle : comment arrivera-on à distribuer des tâches sur plus de 100 millions de coeurs ?
– la nécessité de repenser les intergiciels : MPI restera-il valable à l'échelle exaflopique [5] ?

Mais l'un des principaux défis à relever lors de la conception des machines exaflopiques est le problème des coûts très élevés en termes de consommations d'énergie. Ces coûts très élevés incluant l'installation des infrastructures et les coûts d'entretien deviennent des facteurs prédominants dans le coût total de possession (TCO) d'un superordinateur. La question énergétique à l'échelle exaflopique devient d'autant plus préoccupante lorsque l'on sait que l'on atteint déjà des consommations énergétiques supérieures aux 5 MW à l'échelle pétaflopique alors que le DARPA, le bureau de recherche et de développement du ministère de défense des Etats-Unis d'Amérique, fixe le seuil à 25 MW pour les machines exaflopiques. Les solutions existantes à l'échelle pétaflopique seront-elles valables à l'échelle exaflopique ? Passeront-elles à l'échelle ?

Nos contributions dans cet article consistent à répondre à ces questions d'une part en proposant des adaptations possibles pour les solutions qui existent à l'échelle pétaflopique et d'autre part en suggérant des solutions nouvelles permettant d'aller vers une ère exaflopique efficace en énergie. Nous proposons une architecture verte regroupant les différentes solutions nouvelles que nous avons proposées.

Le reste de l'article s'articule autour du plan suivant. La section 2 expose les diverses propositions de solutions pour répondre à la question énergétique à l'échelle pétaflopique. Dans la section 3, nous tentons d'étudier la validité des solutions énergétiques qui existent déjà et de voir si elles passent à l'échelle. Dans la section 4, nous proposons quelques solutions nouvelles pour que les superordinateurs soient plus être efficaces en énergie. Une dernière section présentera nos conclusions et nos travaux futurs.

2. Etat de l'art

Pour autant que nous sachions, la littérature scientifique ne connaît pour l'instant aucun travail de recherche consistant à apporter des solutions visant à réduire la consommation énergétique dans des machines exaflopiques. En effet, la question de l'efficacité énergétique des plateformes distribuées a été vue essentiellement dans le contexte des grilles de calculs (« grids »), des centres de données (« datacenters »), ou encore des nuages informatiques (« clouds »). Combinées à des optimisations matérielles proposées par les constructeurs, les approches visant à réduire la consommation énergétique des plateformes distribuées peuvent être organisées dans deux classes distinctes.

L'approche « shutdown » consiste à dynamiquement éteindre les ressources inutilisées et à les rallumer seulement quand elles sont nécessaires. De nombreux travaux comme [2, 4] sont basés sur cette approche et suggèrent d'utiliser des algorithmes d'allumages et d'extinctions afin d'éviter que des machines consomment de l'énergie bien qu'elles soient inoccupées. Cependant, pour mettre en oeuvre ces algorithmes d'allumages et d'extinctions, cela nécessite de migrer des tâches d'une machine à l'autre [13] ou de migrer des machines virtuelles sur d'autres noeuds physiques [8, 12]. Ces migrations engendrent également des coûts en terme d'énergie et de performances, dont il est indispensable de tenir compte pour le calcul de l'efficacité énergétique. Cette approche connaît d'autres limites, notamment liées aux pics de consommation que connaît une machine à chaque démarrage. Dans [13], Orgerie et al. proposent de contourner cette limite en prenant en considération ces pics de consommation dans leur approche. Pour cela, ils présentent un modèle de prédiction permettant de prévoir l'utilisation future des res-

sources et définissent une durée minimale à partir de laquelle une ressource consomme moins d'énergie si on l'éteint puis on la rallume plutôt que si on la laisse allumée. Si l'on prédit qu'une ressource sera inutilisée pendant un laps de temps supérieur à cette durée minimale, alors on l'éteint pendant ce laps de temps et on la rallume. Sinon, on la laisse allumée.
L'approche « shutdown » donne des résultats satisfaisants en termes d'économies d'énergie. Cependant, sa mise en oeuvre nécessite la définition d'un modèle de prédiction suffisamment fiable et implique une gestion de la perte de connectivité avec les ressources éteintes.
L'approche « slowdown » consiste à adapter dynamiquement le niveau de performance d'une ressource en fonction du niveau de performance dont on a réellement besoin. Là aussi, de nombreux travaux sont basés sur cette approche et proposent d'utiliser les techniques DVFS (Dynamic Voltage Frequency Scaling) pour les processeurs [9] ou les techniques ALR (Adaptive Link Rate) pour les cartes réseaux [6]. Dans les techniques basées sur DVFS, il est proposé d'adapter le plus justement possible, la vitesse d'horloge du processeur en fonction des performances requises par le CPU. Ces techniques ont inspiré la définition de différents états énergétiques caractérisés par la fréquence du processeur, la tension électrique et la consommation énergétique qui en découle. Les techniques basées sur ALR sont analogues à DVFS, dans la mesure où il s'agit là aussi d'adapter la bande passante du réseau en fonction de l'importance des communications qui ont lieu sur ce réseau. Ces techniques sont de plus en plus prises en compte dans l'implémentation des processeurs d'aujourd'hui notamment avec la norme ACPI[7].
Ces techniques que ce soient celles qui sont basées sur l'approche « shutdown » ou « slowdown » donnent des résultats plus ou moins satisfaisants à l'échelle du pétaflop. Cependant, resteront-elles valables pour des machines exaflopiques ? Faut-il les adapter pour mieux passer à l'échelle ? Ou faut-il prévoir de proposer de nouvelles techniques ou approches visant à réduire la consommation énergétique dans les machines exaflopiques ?

3. Réduire la consommation électrique : du petascale à l'exascale

L'objet de cette partie est de s'interroger sur les solutions énergétiques proposées jusqu'alors et de se demander si elles restent valables à l'échelle exaflopique. Et si ces solutions ne passent pas à l'échelle, alors est-il possible de les adapter pour qu'elles soient toujours valables pour l'informatique exascale ?

3.1. Défis actuels et projections

Un superordinateur est une collection de microprocesseurs fortement connectés entre eux, où chaque microprocesseur se caractérise par une fréquence d'horloge élevée et peut être composé de plusieurs coeurs de calcul. Ainsi, pour concevoir des superordinateurs très performants, les constructeurs jouent simultanément sur le nombre de microprocesseurs, le nombre de coeurs par microprocesseur, la fréquence d'horloge de chaque microprocesseur et en combinant aux microprocesseurs, des processeurs graphiques.
On constate sur la référence LINPACK[8] [3] la machine la plus puissante actuellement, Tianhe-1A, atteint des performances de 2,566 pétaflops et consomme 4,04 MW, tandis que la machine qui la suit dans le classement, Jaguar[9], n'atteint que 1,759 pétaflops pour une consommation de 6,95 MW. Force est de noter que les choix de conception d'un superordinateur peuvent être déterminants pour aboutir à une consommation amoindrie tout en étant plus performant. En effet, Tianhe-1A est 46 % plus performante que Jaguar tout en étant 42 % plus économe en énergie !
Sur le plan architectural, Tianhe-1A combine 7 168 noeuds chacun composé de deux processeurs (CPUs) Intel X5670[10] et un processeur graphique (GPU) Nvidia Tesla M2050[11], soit un total de 14 336 CPUs et 7168 GPUs. Chacun de ces CPUs est capable d'atteindre une performance de 70.32 gigaflops et nécessite une puissance électrique 95W tandis que chaque GPU est capable d'atteindre 515 gigaflops pour 225W. Quant à Jaguar, elle possède 18 688 noeuds chacun composé de deux CPUs AMD Opteron 2435[12], soit un total de 37 376 CPUs. Chacun de ces CPUs est capable d'atteindre 62.4 gigaflops pour une puissance

7. Advanced Configuration and Power Interface : `http://www.acpi.info/DOWNLOADS/ACPIspec40a.pdf`
8. projet LINPACK : `http://www.netlib.org/linpack`
9. Jaguar : `http://www.nccs.gov/jaguar/`
10. `http://ark.intel.com/Product.aspx?id=47920`
11. `http://www.nvidia.com/docs/IO/43395/NV_DS_Tesla_M2050_M2070_Apr10_LowRes.pdf`
12. `http://products.amd.com/en-us/opteroncpuresult.aspx?f1=Six-Core+AMD+Opteron`

électrique de 75W. Quand on essaie d'identifier les différences architecturales de ces deux machines, on peut remarquer que bien que Jaguar soit dotée de plus de microprocesseurs au total, Tianhe-1A est plus performante et plus économe en énergie. Ceci s'explique notamment par l'utilisation de GPUs dans Tianhe-1A étant donné que ces derniers présentent une efficacité énergétique de 2.29 gigaflops/watt à la différence des CPUs utilisés dans Jaguar avec 0.832 gigaflops/watt et des CPUs utilisés dans Tianhe-1A avec 0.74 gigaflops/watt. On ne peut donc s'empêcher de penser que l'usage de processeurs graphiques constitue une piste intéressante dans la conception de superordinateurs verts.
Néanmoins, selon [10], le DARPA suggère de fixer à 25 MW le seuil pour la consommation énergétique des machines exascales, ce qui équivaut à un espoir d'atteindre une efficacité énergétique de 40 gigaflops/watt. Cependant, quand on prend la machine petascale la plus efficace en énergie selon le classement établi par Green 500 [13], on constate qu'elle a une efficacité énergétique inférieure à 1 gigaflop/watt. Les évolutions technologiques actuelles ne suffisent donc plus pour concevoir une machine exascale. À ce niveau, nous pensons qu'il est nécessaire d'envisager de nouvelles pistes de recherche pour être capable d'atteindre ce degré d'efficacité énergétique. Nous proposons en section 4 quelques pistes possibles.

3.2. Techniques pour réduire la consommation énergétique

Comme nous l'avons évoqué dans la section 2, on distingue aujourd'hui deux types d'approches pour réduire la consommation énergétique, à savoir l'approche « shutdown » et l'approche « slowdown ». Néanmoins, ces deux approches sont à adopter avec précaution. Aussi, pour mettre en oeuvre une approche « shutdown », il est souvent indispensable de migrer une ou plusieurs tâches d'un noeud vers un autre de sorte qu'on puisse éteindre le noeud qui n'exécute aucune tâche. Cependant, ces migrations sont gourmandes en énergie et prennent un temps non négligeable, ce qui de surcroît cause inévitablement une dégradation de performances. De telles migrations, plus importantes en nombre et en taille, risquent d'être impraticables à l'échelle exaflopique, d'autant plus que la bande passante du réseau et la mémoire risquent de devenir des ressources encore plus critiques à l'échelle exaflopique [15].
Il faut également souligner le fait que dans un contexte de calcul parallèle à très haute performance, un noeud est bien plus souvent impliqué dans un calcul qu'inoccupé. En d'autres termes, les périodes où les noeuds se retrouvent inoccupés sont tellement courtes, qu'à cause des pics de consommation que l'on connaît à chaque démarrage, on perdrait plus en énergie et en performance en tentant de les éteindre pendant ces périodes d'inactivité. Pour adapter l'approche « shutdown » à l'échelle exaflopique, une première piste de recherche consiste alors à réussir à placer une ressource très rapidement en état d'hibernation (extinction partielle) lors de ces courtes périodes d'inactivité qu'elle connaît.
En ce qui concerne l'approche slowdown, elle reste toujours non optimale dans la mesure où elle ne permet pas d'obtenir une consommation énergétique effectivement proportionnelle au niveau de performance requis à chaque instant. Ce qui explique la non-optimalité de cette approche, c'est le fait que les modèles qui se sont basés sur ces approches (ACPI pour les processeurs, ALR pour le réseau) ont seulement défini quelques états (généralement moins de 6) dans lesquels une ressource (que ce soit un processeur ou une carte réseau) peut se retrouver alternativement au cours de la durée de vie d'une application donnée.
A notre sens, une autre piste de recherche est l'optimisation de cette approche de sorte que l'on réussisse effectivement à minimiser au maximum la consommation énergétique surtout qu'au niveau exascale, le nombre de ressources mises en oeuvre est tellement important que l'on peut moins se permettre de consommer un peu plus que l'on devrait car cela peut aboutir à des pertes énormes à échelle exaflopique. A titre d'exemple, le processeur Intel Xeon 5600 consomme 107 Watts à l'état P3 à une fréquence de 2.8 GHz contre 116 Watts à l'état P2 à 3.0 GHz, ce qui fait une différence de 9 Watts. Or pour atteindre l'échelle exaflopique, le nombre de processeurs Intel Xeon 5600 qu'il faudrait déployer est de l'ordre de 10^6. Par conséquent, une perte de 9 Watts par processeur entraînerait une perte globale de l'ordre de 9 MWatts, ce qui est plus que deux fois la consommation énergétique de Tianhe-1A, la plus puissante machine pétaflopique d'aujourd'hui !
Ainsi pour optimiser l'approche « slowdown », nous proposons de multiplier le nombre d'états énergétiques dans lequel peut se retrouver une ressource de telle sorte que la différence de consommation électrique entre deux états énergétiques successifs soit inférieure à un certain seuil ϵ (0.1W par exemple).

13. Liste Green 500, novembre 2010 : `http://www.green500.org/lists/2010/11/top/list.php`

Ainsi, si on place une ressource donnée dans l'état énergétique e le plus approprié, alors plus ϵ est faible, plus e est proche de l'état énergétique idéal.

4. Vers des solutions nouvelles pour de l'exascale efficace en énergie

Après avoir vu dans la section précédente que les solutions proposées aujourd'hui pour réduire la consommation énergétique des superordinateurs présentent des limites certaines pour l'informatique exaflopique de demain, nous proposons dans la présente section quelques idées de solutions nouvelles en vue d'une consommation efficace en énergie. Force est de remarquer que les solutions jusqu'alors proposées pour être efficace en énergie visent à « consommer moins » d'énergie, c'est-à-dire à réduire la quantité d'énergie consommée. Il faut également chercher des solutions permettant de « consommer mieux », c'est-à-dire à consommer à moindres coûts la même quantité d'énergie et/ou à consommer de l'énergie verte autant que cela est possible. C'est la raison pour laquelle nous proposons dans une première sous-partie des solutions nouvelles pour « consommer moins » d'une part et pour « consommer mieux » d'autre part. Dans une deuxième sous-partie, nous proposons une architecture verte pour les machines exaflopiques reprenant toutes les propositions que nous faisons dans cette section.

4.1. « Consommer moins » vs « consommer mieux »

Nous avons pu relever en fin de section 3.1 que les évolutions technologiques actuelles ne suffisent plus pour concevoir une machine exascale. Pour atteindre une efficacité énergétique aussi importante, une première solution serait de mettre en oeuvre des techniques impliquant une collaboration très fine entre les ressources matérielles et l'application. L'objectif de ces techniques est d'être capable, au niveau applicatif, de prédire de façon intelligente et efficace le niveau de performance requis pour chacune des ressources (processeur et mémoire d'un noeud donné d'une part et bande passante de chaque lien d'autre part) et ce à tout instant.

Ces prévisions sur le niveau de performance requis par ressource sont d'autant plus intéressantes à exploiter que l'on sait que dans une application de calcul parallèle haute performance, on passe successivement dans des phases de :

- calculs intenses, dans lesquelles le processeur est fortement sollicité contrairement à la bande passante et à la mémoire ;
- communications intenses, dans lesquelles la bande passante du réseau est très sollicitée contrairement au processeur et à la mémoire ;
- stockage intense, dans lesquelles la mémoire de stockage local est très sollicitée contrairement au processeur et à la bande passante.

Ce que l'on envisage dans cette solution nouvelle est de se servir d'une couche intermédiaire entre l'application et le matériel qui serait capable d'estimer dynamiquement, à partir d'informations recueillies par analyse du programme (écrit en MPI par exemple), le niveau de performance requis par ressource et le temps d'exécution de chacun des blocs d'instructions dans un calcul parallèle haute performance. En fonction de ces estimations calculées, cette couche intermédiaire sera capable de contrôler les ressources matérielles en les faisant passer d'un état énergétique à un autre et ce dans une optique de se placer dynamiquement dans l'état énergétique le plus approprié aux exigences de l'application, et ce à tout instant.

Une autre façon d'appliquer cette approche nouvelle sans avoir à faire des estimations est d'exécuter l'application une première fois dans le but de recueillir des informations sur l'évolution des exigences en performances pour chacune des ressources tout au long de l'application. Et pour toutes les fois suivantes où l'on exécute cette application exascale, ces informations seront utilisées pour permettre à chacune des ressources mises en jeu de se placer à tout instant dans l'état énergétique le plus bas qui puisse satisfaire les exigences en performances.

Pour « consommer moins », une autre solution est d'envisager de demander et de prendre en compte les exigences des utilisateurs des superordinateurs en termes de performances. En effet, l'obtention des meilleurs temps d'exécution ne devrait pas être la seule finalité d'un utilisateur de machine exaflopique. A notre sens, il faut préférer à cela l'obtention de la plus basse consommation énergétique qui satisfait les exigences minimales des utilisateurs en termes de performances. N'est-il pas mieux d'arriver à temps voulu et à moindres coûts plutôt que d'arriver le plus vite possible ?

Afin d'illustrer cette idée, considérons une application qui dans le meilleur des cas s'exécute en 10 secondes sur une plateforme atteignant l'échelle exaflopique. Si un utilisateur nous informe qu'il souhaiterait que son application s'exécute en moins de 2 minutes, il est ainsi possible de jouer sur le nombre de ressources nécessaires pour que cette application tourne en moins de 2 minutes. En impliquant moins de ressources pendant une période plus longue, on réduit la consommation énergétique car on évite ainsi tous les pics de consommation (dus aux démarrages) qu'auraient connus les ressources que l'on a choisi de ne pas mettre en jeu et donc de ne pas allumer.
La conception des applications de calcul haute performance de demain devrait donc intégrer ces exigences en termes de performances de sorte à les respecter tout en optimisant l'énergie nécessaire.
Afin de « consommer mieux », nous préconisons aux utilisateurs et aux administrateurs des superordinateurs d'aménager pendant les périodes creuses les applications exigeant les plus fortes consommations énergétiques de sorte que l'énergie leur coûte moins cher et de sorte que l'énergie consommée soit plus verte quand cela est possible. Pour cela, les utilisateurs pourront également fixer des contraintes sur la planification des exécutions de leurs applications de sorte que les administrateurs puissent aménager ces exécutions pendant les périodes creuses. En effet, les fournisseurs d'énergie proposent généralement des prix pour le kWh en fonction de l'heure et le jour où l'énergie est consommée. À titre d'exemple, dans [14], EDF présente les prix en euros du kWh selon que l'on soit un jour bleu, un jour blanc ou un jour rouge, mais aussi selon que l'on soit en heure creuse (de 22h à 6h) ou en heure pleine (de 6h à 22h). Force est de constater que le prix du kWh en jours rouges et en heures pleines est plus que 7 fois le prix du kWh en jours bleus et en heures creuses, ce qui est loin d'être négligeable ! Pourquoi ne pas en profiter pour exécuter ses applications de calcul haute performance en périodes creuses ?
Au-delà de cette première recommandation, nous proposons de mettre en oeuvre une « smart grid » (un réseau de distribution d'électricité « intelligent ») permettant d'établir une meilleure communication entre les fournisseurs d'énergie et leurs plus importants clients, ce qui peut-être à la fois profitable pour les fournisseurs d'énergie tel qu'EDF et pour les clients les plus consommateurs, et spécialement ceux qui feront fonctionner des applications exaflopiques. En effet, si on conçoit des applications de calcul haute performance capables d'envoyer des informations sur les prévisions de consommations énergétiques aux fournisseurs d'énergies via des flux permanents de communication, alors ces derniers peuvent plus aisément s'approvisionner en énergie verte et ainsi la proposer à des tarifs amoindris.

4.2. Vers une architecture verte pour les machines exaflopiques

L'architecture que nous proposons reprend les diverses suggestions que nous avons faites dans les deux sous-sections précédentes. Elle repose aussi bien sur des interactions internes au sein de chaque noeud de la machine exaflopique que sur des interactions externes mettant en jeu différents acteurs.
Ainsi sur la figure 1, sont représentées les différentes interactions internes que nous prévoyons entre les couches application et matériel de chaque noeud et ce par le biais d'une couche intermédiaire pour la gestion des ressources. La couche application fournit à la couche de gestion des ressources, des estimations sur les exigences en termes de performances pour chaque ressource (processeur, mémoire, réseau). Grâce à ces estimations, la couche de gestion des ressources place chacune des ressources dans l'état énergétique optimal, c'est-à-dire le plus bas état qui permet d'atteindre les performances requises pour chaque ressource.

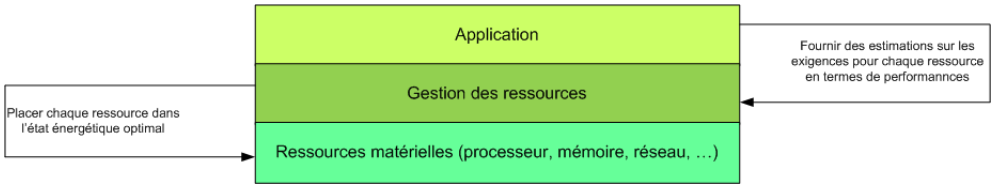

FIGURE 1 – Architecture en couches de chaque noeud de la machine exaflopique

14. EDF, les prix de l'électricité : http://bleuciel.edf.com/abonnement-et-contrat/les-prix/les-prix-de-l-electricite/option-tempo/en-savoir-plus-52429.html

À titre d'exemple, si dans un code MPI on rencontre un bloc d'instructions où des MPI_Bcast, MPI_Recv, MPI_Send se succèdent, on peut donc en déduire que l'on se trouve dans une phase de communications intenses dans laquelle le réseau est la ressource la plus sollicitée. Il s'agit d'être capable d'estimer la bande passante nécessaire pour la bonne exécution de ce bloc d'instructions mais aussi d'évaluer la fréquence d'horloge du noeud qui exécute ce bloc d'instructions ainsi que l'importance (en nombre et en taille) des accès à la mémoire indispensables. Une fois estimées, ces valeurs permettent à la couche de gestion de ressources de dimensionner et de placer chaque ressource dans l'état énergétique optimal en termes de fréquence pour le processeur, de bande passante pour le réseau, ou de la taille mémoire.
Sur la figure 2, sont représentées les différentes interactions externes entre la machine exaflopique, son utilisateur (celui qui souhaite exécuter une application exaflopique), son administrateur, et le fournisseur d'énergie (par exemple EDF en France). Avant de lancer une application exafloplique, l'utilisateur précise les jours qui conviendraient à l'exécution de son application ainsi que ses exigences en termes de performances et de coûts financiers en fournissant le temps d'exécution maximal et le prix en euros qu'il ne voudra pas dépasser à chaque lancement de l'application. En tenant compte de ces informations, le gestionnaire de ressources privilégie autant que possible les périodes creuses et choisit de mettre en jeu le minimum de ressources possibles permettant d'atteindre les performances exigées par l'utilisateur. Si malgré cela l'application atteint des pics de consommation trop importants, le fournisseur d'énergie est prévenu afin qu'il puisse fournir au moment voulu suffisamment d'énergie et éventuellement une énergie verte.

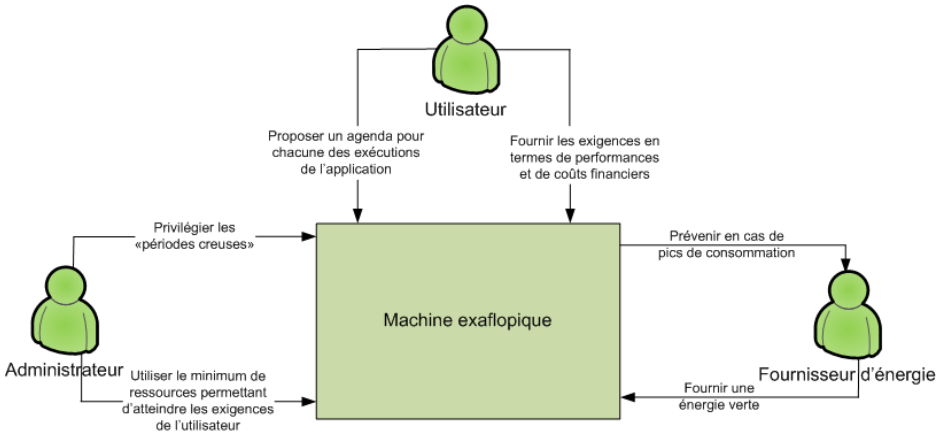

FIGURE 2 – Interactions externes responsabilisant les différents acteurs extérieurs

En somme, il s'agit d'une double négociation d'une part entre l'utilisateur et l'administrateur et d'autre part entre l'administrateur et le fournisseur d'énergie. Au niveau de la négociation utilisateur - administrateur, il s'agit de trouver un compromis qui satisfasse à la fois l'utilisateur qui désire atteindre les meilleures performances au moindre coût financier et l'administrateur qui facture à l'utilisateur les coûts de la consommation énergétique qui varient en fonction de la période d'utilisation et de l'importance des ressources mises en jeu. En ce qui concerne la relation administrateur - fournisseur d'énergie, il s'agit de rendre service au fournisseur d'énergie en le prévenant des pics de consommation afin qu'il puisse réguler son approvisionnement, ce qui légitime le fait que l'administrateur négocie des tarifs préférentiels et permet éventuellement au fournisseur d'énergie de délivrer une énergie plus verte.

5. Conclusion

Dans cet article, nous avons traité la question de l'efficacité énergétique dans les machines exaflopiques. Pour cela, nous avons étudié les solutions existantes et envisagé la possibilité de les adapter de sorte qu'elles puissent rester valables à l'échelle exaflopique. Nous avons ensuite proposé une architecture

verte pour les machines exaflopiques regroupant les différentes solutions nouvelles visant à « consommer moins » d'énergie et à « consommer mieux ». Dans cette architecture, nous avons prévu aussi bien des interactions internes au sein de chaque noeud de la machine que des interactions externes avec les différents acteurs intervenant directement ou indirectement sur la machine. D'une part, nous avons préconisé une collaboration plus fine entre l'application et les ressources matérielles dans une optique de réduire la consommation énergétique. D'autre part, nous avons suggéré une coopération entre l'utilisateur de la machine, le gestionnaire de ressources et le fournisseur d'énergie dans un but de « consommer mieux ». Ces propositions constituent une première étape sur le long chemin nécessaire à la conception des machines exaflopiques vertes. Dans nos travaux futurs, nous projetons de nous appuyer sur cette architecture verte pour proposer et évaluer des algorithmes efficaces pour mettre en oeuvre nos propositions à l'échelle exaflopique.

Par ailleurs selon [15], les processeurs représentent une part importante (56%) dans la consommation globale des systèmes actuels de calcul haute performance contrairement à la mémoire (9%) et le réseau (33%). Cependant, cette répartition risque de basculer en ce qui concerne les machines exaflopiques vu l'importance du réseau et de la mémoire dans de tels systèmes. Pour cette raison, nous projetons aussi de nous pencher essentiellement sur des approches en faveur d'une réduction de la consommation de la part réseau et mémoire dans de tels systèmes.

Bibliographie

1. Franck Cappello, Al Geist, Bill Gropp, Sanjay Kale, Bill Kramer, et Marc Snir. Toward exascale resilience. *International Journal of High Performance Computing Applications*, 23 :374–388, November 2009.
2. Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, et Ronald P. Doyle. Managing energy and server resources in hosting centers. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP'01, pages 103–116, Banff, Alberta, Canada, October 2001. ACM.
3. Jack Dongarra. The LINPACK Benchmark : An Explanation. In *ICS : Proceedings of 1st International Conference Supercomputing, Athens, Greece, June 8-12, 1987*, volume 297 of *Lecture Notes in Computer Science*. Springer, 1987.
4. Xiaobo Fan, Wolf-Dietrich Weber, et Luiz Andre Barroso. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th annual international symposium on Computer architecture*, ISCA '07, pages 13–23, New York, NY, USA, 2007. ACM.
5. William Gropp. MPI at Exascale : Challenges for Data Structures and Algorithms. In *Proceedings in Recent Advances in Parallel Virtual Machine and Message Passing Interface, 16th European PVM/MPI Users' Group Meeting, September 7-10, 2009, Espoo, Finland.*, volume 5759 of *Lecture Notes in Computer Science*, page 3. Springer, 2009.
6. Chamara Gunaratne et Kenneth J. Christensen. Ethernet adaptive link rate : System design and performance evaluation. In *The 31st IEEE Conference on Local Computer Networks, Tampa, Florida, USA, 14-16 November 2006*. IEEE Computer Society, 2006.
7. J. Hack et E. Bierly. Computational and informational technology rate limiters to the advancement of climate change science. DOE Advanced Scientific Computing Research Advisory Committee, November 6-7 2007.
8. Fabien Hermenier, Nicolas Loriant, et Jean-Marc Menaud. Power Management in Grid Computing with Xen. In *Frontiers of High Performance Computing and Networking - ISPA 2006 Workshops, ISPA 2006 International Workshops, FHPCN, XHPC, S-GRACE, GridGIS, HPC-GTP, PDCE, ParDMCom, WOMP, ISDF, and UPW*, volume 4331 of *Lecture Notes in Computer Science*, pages 407–416, Sorrento, Italy, December 4-7 2006. Springer.
9. Y. Hotta, M. Sato, H. Kimura, S. Matsuoka, T. Boku, et D. Takahashi. Profile-based optimization of power performance by using dynamic voltage scaling on a pc cluster. In *Proceedings of the 20th International in Parallel and Distributed Processing Symposium, IPDPS 2006*, 2006.
10. P. M. Kogge et et al. ExaScale Computing Study : Technology Challenges in Achieving Exascale Systems. In *DARPA Information Processing Techniques Office*, page pp. 278, Washington, DC, September 28 2008.
11. Douglas B. Kothe. Science Prospects and Benefits with Exascale Computing. Technical Report ORNL/TM-2007/232, OAK Ridge National Laboratory, December 2007.
12. Anne-Cécile Orgerie et Laurent Lefèvre. When clouds become green : the green open cloud architecture. In *Parco2009 : International Conference on Parallel Computing*, Lyon, France, September 2009.
13. Anne-Cecile Orgerie, Laurent Lefevre, et Jean-Patrick Gelas. Save Watts in your Grid : Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems. In *ICPADS 2008 : The 14th IEEE International Conference on Parallel and Distributed Systems, Melbourne, Australia*, December 2008.
14. Thomas Spelce. Sequoia and the petascale era. Presentation at SCICOMP 15, May 20 2009.
15. Thomas Sterling. An overview of exascale architecture challenges. SC08 Workshop on the Path to Exascale, November 16 2008.